AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1-39.   (Cancelled).

40.   (New) A method for facilitating extension of application functionality by generating one or more application programming interfaces (API) that facilitate document development using domain terminology rather than native terminology of a host application, the method comprising:

using a processor of a computing device:

accessing a schema component that is stored on one or more computer-readable storage media, the schema component including a schema element representative of at least one domain terminology term of one or more problems for solving in a host application, wherein the at least one domain terminology term is different from native terminology utilized in a general API of the host application, wherein the native terminology identifies elements by at least one of address or range, and the domain terminology identifies the same elements with textual descriptive terms;

using a mapping component stored on the one or more computer-readable storage media in constructing an API of the host application, such that domain terminology that includes the textual descriptive terms maps to native terminology that includes at least one of address or range information, and thereby enabling the host application to operate on the domain terminology that utilizes textual descriptive terms; and

using a generating component to produce a new API based upon a mapping of the mapping component, wherein using the generating components provides a developer with document development using the textual descriptive terms in the host application, and in lieu of the address or range information of the native terminology of the host application.

41.   (New) The method of claim 40, wherein the at least one domain terminology term is accessed from a generic API with terminology different than the native terminology of the host application.

42.   (New) The method of claim 40, wherein the at least one domain terminology term includes a plurality of different domain terminology terms, wherein the plurality of different domain terminology terms are textual descriptive terms from multiple different generic APIs.

43.   (New) The method of claim 40, further comprising using a separation component to separate data from document content.

44.   (New) The method of claim 40, further comprising:

using a separation component to generate a data island in a document of the host application;

making the data island in the document of the host application editable without launching of the host application; and

synchronizing contents of the data island and the document, such that the document is updated with modified information of the data island when the document is launched within the host application.

45.   (New) The method of claim 40, wherein the schema component further facilitates manipulation of a variable without reference to underlying register and/or stack allocations.

46.   (New) One or more computer-readable storage media having stored thereon computer-executable instructions that, when executed by one or more processors of a computing system, perform a method for facilitating extension of application functionality by generating one or more application programming interfaces (API) that facilitate document development using domain terminology rather than native terminology of a host application, the method comprising:

accessing a schema component, the schema component including a schema element representative of a at least one domain terminology term of one or more problems for solving in a host application, wherein the at least one domain terminology term is different from native terminology utilized in a general API of the host application, wherein the native terminology identifies elements by at least one of address or range, and the domain terminology identifies the same elements with textual descriptive terms;

using a mapping component in constructing an API of the host application, such that domain terminology that includes the textual descriptive terms maps to native terminology that includes at least one of address or range information, and thereby enabling the host application to operate on the domain terminology that utilizes textual descriptive terms; and

using a generating component to produce a new API based upon a mapping of the mapping component, wherein using the generating components provides a developer with document development using the textual descriptive terms in the host application, and in lieu of the address or range information of the native terminology of the host application.

47.   (New) The one or more computer-readable storage media of claim 46, wherein the at least one domain terminology term is accessed from a generic API with terminology different than the native terminology of the host application.

48.   (New) The one or more computer-readable storage media of claim 46, wherein the at least one domain terminology term includes a plurality of different domain terminology terms, wherein the plurality of different domain terminology terms are textual descriptive terms from multiple different generic APIs.

49.   (New) The one or more computer-readable storage media of claim 46, wherein the method further comprises using a separation component to separate data from document content.

50.   (New) The one or more computer-readable storage media of claim 46, wherein the method further comprises:

using a separation component to generate a data island in a document of the host application;

making the data island in the document of the host application editable without launching of the host application; and

synchronizing contents of the data island and the document, such that the document is updated with modified information of the data island when the document is launched within the host application.

51.   (New) The one or more computer-readable storage media of claim 46, wherein the schema component further facilitates manipulation of a variable without reference to underlying register and/or stack allocations.

52.   (New) A computing system for facilitating extension of application functionality by generating one or more application programming interfaces (API) that facilitate document development using domain terminology rather than native terminology of a host application, the computing system comprising:

one or more processors; and

one or more computer-readable media having stored thereon computer-executable instructions that, when executed by the one or more processors, cause the computing system to:

access a schema component the schema component including a schema element representative of a at least one domain terminology term of one or more problems for solving in a host application, wherein the at least one domain terminology term is specific to a particular document and is different from native terminology utilized in a general API of the host application, wherein the native terminology identifies elements by at least one of address or range, and the domain terminology identifies the same elements with textual descriptive terms;

use a mapping component stored on the one or more computer-readable storage media in constructing an API of the host application, such that domain terminology that includes the textual descriptive terms maps to native terminology that includes at least one of address or range information, and thereby enabling the host application to operate on the domain terminology that utilizes textual descriptive terms; and

use a generating component to produce a new API based upon a mapping of the mapping component, wherein using the generating components provides a developer with document development using the textual descriptive terms in the host application, and in lieu of the address or range information of the native terminology of the host application.

53.   (New) The computing system of claim 52, wherein the at least one domain terminology term is accessed from a generic API with terminology different than the native terminology of the host application.

54.   (New) The computing system of claim 52, wherein the at least one domain terminology term includes a plurality of different domain terminology terms each specific to particular document, wherein the plurality of different domain terminology terms are textual descriptive terms from multiple different generic APIs.

55.   (New) The computing system of claim 52, wherein executing the computer-executable instructions further causes the computing system to use a separation component to separate data from document content.

56.   (New) The computing system of claim 52, wherein executing the computer-executable instructions further causes the computing system to:

use a separation component to generate a data island in a document of the host application;

make the data island in the document of the host application editable without launching of the host application; and

synchronize contents of the data island and the document, such that the document is updated with modified information of the data island when the document is launched within the host application.

57.   (New) The computing system of claim 56, wherein the schema component further facilitates manipulation of a variable without reference to underlying register and/or stack allocations.

58.   (New) The computing system of claim 57, wherein:

the generation component is further used to generate a data programming model and a view programming model that are automatically connected to each other via data binding, the view programming model providing an interface by which the host application operates on the domain terminology;

generating from the schema element, which is a user-defined schema element, code having a form x.y, wherein "x" is a portion generated from the user-defined schema element, "y" is a property provided by the host application, and "." is a period separating "x" and "y"; and

the host application is a spreadsheet that references cell content using cell addresses and cell ranges in its native terminology, and accesses the same cell content using textual heading information using the domain terminology.